

**С.М. ГРИША**, д-р. техн. наук, НТУУ «КПІ», Київ

**О.С. РОДІЧЕВА**, НТУУ «КПІ», Київ

**Д.І. ПРИЛПКО**, НТУУ «КПІ», Київ

## ТЕХНОЛОГІЧНО ІНТЕЛЕКТУАЛІЗОВАНІ ІНФОРМАЦІЙНІ СИСТЕМИ ДЛЯ УПРАВЛІННЯ БІЗНЕСОМ (ІСУБ) НА ОСНОВІ АЛГЕБРИ ПОКАЗНИКІВ

В статті розглядається оригінальна технологія побудови прикладного програмного забезпечення (ППЗ) систем управління бізнесом, яка дозволяє створювати та оперативно змінювати ППЗ без залучення програмістів та ІТ-технологів шляхом автоматичного синтезу та оптимізації коду ППЗ та структури бази даних (БД) за допомогою спеціальної технологічної бази знань (БЗ).

Article describes original technology of application software (software) construction of business management systems, which allows to create and operatively change software without involving programmers and IT-technologists by an automatic synthesis and code optimization and database (DB) structure optimization using the special technological knowledge base (KB).

**1. Архітектура ІСУБ та проблема ефективності.** При оцінці ефективності ІСУБ зараз покладаються на показники вартості та швидкості створення, розвитку, адаптації до зміни середовища. Ці показники залежать від вибору архітектури ІСУБ та концепції операцій при створенні, розвитку та адаптації.

Системна модель архітектури сучасних ІСУБ має вигляд сімки

$$\langle D_1, D_2, S_{1 \rightarrow 2}, T, Z_{d \rightarrow t}, Z_{t \rightarrow d}, Z_{t \rightarrow t} \rangle, \quad (1)$$

Де:  $D_1, D_2$  – вхідні (первинні) та вихідні (похідні) документи системи, семантику яких задають користувачі чи експерти із змісту бізнесу та менеджменту;

$S_{1 \rightarrow 2}$  – семантичний опис бізнес-правил отримання похідних документів із первинних чи інших похідних, джерело якого аналогічне попередньому;

$T$  – таблиці БД, які задаються інформаційними технологіями шляхом відображення  $D, S$  на  $T$  виходячи з міркувань «фізики» обробки [1];

$Z_{d \rightarrow t}, Z_{t \rightarrow d}, Z_{t \rightarrow t}$  – запити на маніпуляції з даними в БД відповідно у випадку внесення користувачем змін у первинні документи, формування

похідних документів та формування матеріалізованих віртуальних подань [1].

Характерним недоліком моделі (1) є еkleктичність елементів  $D_1, D_2$ , які є продуктом суб'єктивних поглядів користувачів на зручність отримання інформації для управління бізнесом. Ця еkleктичність полягає в тому, що документ – це довільна колекція певних економічних чи технічних показників [6], яка може змінюватись і не має суттєво впливу на філософію управління.

В наслідок цього пропонується модель (2), доведення права на існування та перспективності якої є однією з цілей статті.

$$\langle P_1, P_2, D_1 = A(P_1), D_2 = A(P_2), S_{P_1 \rightarrow P_2}, T, Z_{d \rightarrow t}, Z_{t \rightarrow d}, Z_{t \rightarrow t} \rangle \quad (2)$$

Модель (2) відрізняється тим, що користувач спочатку описує простір вхідних та похідних показників  $P_1, P_2$ , документи задаються у формі виразів алгебри показників  $D_1 = A(P_1), D_2 = A(P_2)$ , семантика оброблення даних задається в термінах показників  $S_{P_1 \rightarrow P_2}$  і це, як буде показано далі, дає можливість не тільки формалізувати діяльність інформаційних технологів, але й здійснити ефективне відображення бізнес-правил на технологічну підсистему, тобто структуру БД та прикладне програмне забезпечення. Це дає нам право говорити про інтелектуалізованість запропонованої технології.

**2. Технологічна та функціональна інтелектуалізованість.** Інтелектуалізованою (ІС) назвемо системи, побудовані на основі інженерії знань. Знання формалізуються у формі правил та фактів, а для обробки застосовуються абстрактні механізми висновків, що суттєво поліпшує гнучкість і внаслідок цього застосовуваність, а з нею і економічну конкурентноздатність ІТ.

Сучасні ІС є переважно функціонально інтелектуалізованими, тобто вони мають справу із знаннями про функціонування об'єкту управління, в даному випадку, бізнес системи. Але успіху (розповсюдженню та проникненню в практику) заважає, серед іншого, не дуже висока ефективність оброблення знань, недостатній рівень їх формалізації, дуже великий обсяг (більше сотень тисяч фактів) та високий темп реального часу оперативного контуру управління (секунди, хвилини, години).

Що стосується інформаційно-технологічної складової знань, то перше, вони мають значно менший обсяг, по-друге, більш високий рівень формалізації і, по-третє, ці знання можливо обробляти не в оперативному

контурі системи, а в стратегічному, для якого темп реального часу є значно повільнішим (дні, тижні, місяці).

### 3. Концептуальна модель операцій при створенні ІСУБ із застосуванням існуючих технологій. Ця модель має такий вигляд.

- 3.1. Користувачі описують  $D_1, D_2, S_{1 \rightarrow 2}$  моделі (1);
  - 3.2. Інформаційні технологи розробляють структуру БД ( $T$ );
  - 3.3. Аналітики роблять постановки задач для програмістів спираючись на побудовану БД;
  - 3.4. Програмісти розробляють ППЗ у формі компонентів на мовах програмування високого рівня або у формі скриптів для компонентів, що управляються мета даними;
  - 3.5. Аналітики здійснюють тестування семантики ППЗ і після виправлення помилок програмістами воно приймається за критерієм коректності;
  - 3.6. ІТ тестують ППЗ за критерієм швидкодії і при необхідності ставлять задачу програміста щодо внесення змін в ППЗ та БД, після чого попередні пункти повторюються до повного задоволення вимог або їх пом'якшення;
  - 3.7. Система остаточно приймається і впроваджується, але при потребі її розвитку та адаптації до змін середовища все повторюється а якщо це не можливо зробити оперативно, то система на певний час функціонально деградує.
- Недоліки такої схеми операцій є причиною великої вартості систем та їх адаптацій. Прогрес відбувається без зміни схеми, а виключно в сфері інструментальних засобів а також у можливостях стандартних ERP.

### 4. Концептуальна модель операцій при створенні технологічно інтелектуалізованих ІСУБ. Ця модель має такий вигляд.

- 4.1. Користувачі описують  $P_1, P_2, S_{P1 \rightarrow P2}$  моделі (2).
- 4.2. Аналіз цілісності під моделі  $P_1, P_2, S_{P1 \rightarrow P2}$  здійснюється автоматично і при необхідності аналітики пояснюють користувачам суть проблеми;
- 4.3. Користувачі описують  $D_1, D_2$  моделі (2);
- 4.4. Інформаційні технологи лише при необхідності уточнюють технологічну БЗ, структура якої буде розглянута далі;
- 4.5. Автоматично генерується БД та ППЗ;

4.6. Програмісти розробляють ППЗ у формі компонентів на мовах програмування високого рівня лише для обмеженої функціональності із врахуванням 4.4;

4.7. Аналітики здійснюють тестування семантики ППЗ виключної частини функціональності і після виправлення помилок програмістами воно приймається за критерієм коректності;

4.8. Відповідність ППЗ за критерієм швидкодії забезпечується оптимізаційними правилами технологічної БЗ;

4.9. Система остаточно приймається і впроваджується, і при потребі її розвитку та адаптації до змін середовища здійснюється часткове повторення описаних операцій.

Перевага такої схеми операцій полягає у зменшенні кількості потенційно похибно-небезпечних комунікацій між спеціалістами та в цілому зменшенні залежності від суб'єктивного фактору, яким виступають ІТ-спеціалісти. У порівнянні із застосуванням стандартних ERP суттєво збільшується рівень адаптованості ІТ до потреб конкретного бізнесу, крім того відомим є досить висока трудомісткість адаптації таких систем, як значні втрати часу на адаптацію, проте ці системи виконують роль акумулятора прогресивного управлінського досвіду.

### 5. Базові властивості алгебри показників. Показники варто використати в якості головного формалізму систем даного роду з таких причин:

- В термінах показників формулюється бачення економістів та бізнесменів про нормативний процес управління;
- Показники є узагальненням лінійної алгебри, а вона є конструктивним апаратом для планових розрахунків;
- Властивості алгебри показників базуються на властивостях реляційної алгебри, а вони глибоко опрацьовані в роботах Кодда [2].

Треба також брати до уваги, що документи (основні інформаційні інструменти управління бізнесом) представляють собою не що інше, як колекції показників у формі їх «природного сполучення» [1].

Таблиці БД, отримані в результаті нормалізації [1] також є природними сполученнями показників, а показники – це елементарні відношення, що знаходяться щонайменше в третій нормальній формі [1], тобто не підлягають нормалізації.

Треба також брати до уваги, що при описі ПО у формі показників процес побудови нормалізованої БД відбувається за синтетичним сценарієм [3], а не за декомпозиційним [1], а отже відпадає необхідність розв'язання багатьох комбінаторних задач [1].

На відміну від [3], вважати, що синтез реалізується не тільки за рахунок природного сполучення, але й об'єднання, як показано у наступних правилах (3,4).

$$x = \triangleright \triangleleft x_i \Rightarrow x_i = \pi_{sh(x_i)}(x), \quad (3)$$

$$x = \bigcup_{i \in I} x_i \Rightarrow x_i = \sigma_{f(x_i)}(x), \quad (4)$$

де  $f(x_i)$  – обмеження цілісності для показника  $x_i$ .

**6. Правила висновків для технологічних метаданих.** Роль правил у запропонованій теорії полягає у перекладі описаних показників на мову програмування та формування таблиць БД. Критеріями доцільності побудованої ІСУБ є:

- Відповідність оперативності масштабу реального часу;
- Мінімальний обсяг обчислювальної роботи системи в період пікового навантаження.

Беручи до уваги те, що гнучкість системи та відповідно витрати із супроводу є продуктом застосування інтелектуалізації, врахування зазначених критеріїв забезпечує мінімізацію вартості володіння ІТ.

Якщо не враховувати час реакції системи та обсяг роботи, тобто вважати що всі проблеми вирішуються за рахунок потужності сервера, що у великій кількості випадків цілком допустимо, то діють правила організації показників в таблиці (5,6) а правила отримання документів визначаються за допомогою (3,4).

$$\forall i (shr(x_i) = k) \Rightarrow \exists (T = \triangleright \triangleleft x_i), \quad (5)$$

$$\forall i (sh(x_i) = k) \Rightarrow \exists (T = \bigcup_{i \in I} x_i), \quad (6)$$

де

$shr(x_i)$  – реляційна складова показника  $x_i$  (ключ),

$sh(x_i)$  – схема відношення  $x_i$ .

Якщо враховувати тільки критерій мінімізації обсягу роботи, тобто вважати що тільки проблеми оперативності вирішуються за рахунок потужності сервера, то діють складніші правила організації показників в таблиці (7,8), а документи визначаються за допомогою правил (3,4).

$$\begin{aligned} shr(x_i) = shr(x_j) \wedge \exists d (x_i = \pi_{sh(x_i)}(d) \wedge x_j = \pi_{sh(x_j)}(d)) \Rightarrow \\ \exists T (x_i = \pi_{sh(x_i)}(T) \wedge x_j = \pi_{sh(x_j)}(T)) \end{aligned} \quad (7)$$

Тобто, якщо існує документ, в якому показники з однаковими реляційними складовими з'єднані, то має існувати таблиця БД, в якій вони з'єднані, в іншому випадку обсяг обчислень системи зростає на величину необхідних з'єднань.

$$\begin{aligned} sh(x_i) = sh(x_j) \wedge \\ \exists d (x_i = \sigma_{f(x_i)}(\pi_{sh(x_i)}(d)) \wedge x_j = x_i = \sigma_{f(x_j)}(\pi_{sh(x_j)}(d))) \Rightarrow \\ \exists T (x_i = \sigma_{f(x_i)}(\pi_{sh(x_i)}(T)) \wedge x_j = x_i = \sigma_{f(x_j)}(\pi_{sh(x_j)}(T))) \end{aligned} \quad (8)$$

Тобто, якщо існує документ, в якому показники з однаковими схемами об'єднані, то має існувати таблиця БД, в якій вони об'єднані, в іншому випадку обсяг обчислень системи зростає на величину необхідних об'єднань.

Зрозуміло, що нас має цікавити мінімальна множина таблиць, які задовольняють умовам (7,8). Її можна отримати шляхом поглинання.

Нетривіальною є проблема спрощення алгебраїчних виразів в наслідок того, що правила еквівалентних перетворень алгебраїчних виразів, досліджені та описані в [1], орієнтовані виключно на проблеми зменшення реактивності запитів. При цьому задача спрощення виразів судячи з аналізу публікацій ще не ставилась.

Розглянемо технологію спрощення виразів на основі поглинання при природному сполученні, яка є дуже важливою у контексті теми, оскільки при прямому виконанні операцій з показниками, які є фактично проекціями таблиць БД довелося би виконувати багато зайвих операцій.

$$\begin{aligned} x_i = \pi_s(\sigma_f(T)) \\ \pi_{S_1}(T_1) \triangleright \triangleleft \pi_{S_2}(T_2) = \pi_{S_1 \cup S_2}(T_1 \triangleright \triangleleft T_2) \end{aligned} \quad (9)$$

$$\begin{aligned} T_1 = T_2 = T \\ \pi_{S_1}(T) \triangleright \triangleleft \pi_{S_2}(T) = \pi_{S_1 \cup S_2}(T \triangleright \triangleleft T) \\ T \triangleright \triangleleft T = T \\ \pi_{S_1}(T) \triangleright \triangleleft \pi_{S_2}(T) = \pi_{S_1 \cup S_2}(T) \end{aligned} \quad (10)$$

В результаті спрощення дві проекції та природне сполучення в лівій частині (9) замінено на одну проекцію в (10).

Аналогічні правила спрощення можливо побудувати для операції об'єднання  $\cup$  та селекції по ортогональним умовам .

$$\begin{aligned}\sigma_{f_1}(T_1) \cup \sigma_{f_2}(T_2) &= \sigma_{f_1 \vee f_2}(T_1 \cup T_2) \\ T_1 &= T_2 = T \\ \sigma_{f_1}(T_1) \cup \sigma_{f_2}(T_2) &= \sigma_{f_1 \vee f_2}(T \cup T) \\ T \cup T &= T \\ \sigma_{f_1}(T) \cup \sigma_{f_2}(T) &= \sigma_{f_1 \vee f_2}(T)\end{aligned}$$

**7. Приклад опису показників, документів та побудови БД та ППЗ.** Для прикладу візьмемо спрощений опис виробництва продукції. Нехай основними показниками виробництва є норма виробництва («норма»), та кількість виробленої продукції («вироблено»). Ці показники задаються на відношеннях атрибутів цех, тип продукції, період виробництва, які мають відповідні домени.

Із описаних атрибутів можна утворити два показники :

- «норма» ( «цех», «тип продукції», «період виробництва»)
- «вироблено» («цех», «тип продукції», «період виробництва» )

Нехай ці два показники є первинними показниками, тобто їх не можна вивести із інших, а треба вводити в систему, як відображення подій на об'єкті. Ці показники зручно вводити через документи «Нормативи виробництва» та «Виробництво продукції». Документи мають додаткову інформацію – дату документу, код та коментар. Також потрібен звіт, в якому відображається нормативи для виробленої продукції та співвідношення нормативного виробництва (ще один показник «продуктивність») до фактичного для оцінки продуктивності виробництва за певний період.

Відповідно до п.п. 4.1. описуємо модель (2). Для цього необхідно задати  $P_1, P_2$ .

Вхідні показники:

$$P_1 = \{ \text{"норма", "виробництво"} \} = \{x_1, x_2\}$$

$$sh(x_1) = \{ \text{"цех", "тип продукції", "період виробництва", "кількість", "одиниця виміру"} \} = \{a_1, a_2, a_3, a_4, a_5\}$$

$$sh(x_2) = \{ \text{"цех", "тип продукції", "період виробництва", "кількість", "одиниця виміру"} \} = \{a_1, a_2, a_3, a_4', a_5\}$$

Тут «кількість» позначено літерою зі штрихом. Цим позначено те, що кількість така ж сама як і в показнику «норма», але використовується в ролі виробленої кількості, а не планової.

Вихідні показники:

$$P_2 = \{ \text{"продуктивність"} \} = \{x_3\}$$

$$sh(x_3) = \{ \text{"цех", "тип продукції", "період виробництва", "кількість планова", "кількість факт.", "одиниця виміру"},$$

$$\text{"виконання плану у відсотках"} \} = \{a_1, a_2, a_3, a_4, a_4', a_5, a_6\}$$

Задамо  $S_{P_1 \rightarrow P_2}$  :

$$a_6 = a_4 / a_4'$$

Далі відповідно до п.п. 4.2 система проведе аналіз цілісності опису показників. Після цього, відповідно до п.п. 4.3. користувач описує  $D_1, D_2$  моделі. В алгебрі показників це буде виглядати наступним чином:

Вхідні (первинні) документи системи

$$D_1 = \{ \text{"Нормативи виробництва", "Виробництво продукції"} \} = \{d_1, d_2\}$$

$$d_1 = x_1$$

$$d_2 = x_2$$

$$shr(x_1) = \{ \text{"цех", "тип продукції", "період виробництва"} \} = \{a_1, a_2, a_3\}$$

$$shr(x_2) = \{ \text{"цех", "тип продукції", "період виробництва"} \} = \{a_1, a_2, a_3\}$$

У всіх документів мають бути системні атрибути такі як код документу, дата та назва. Але так як вони у всіх документів присутні з розгляду в прикладі можна опустити, зазначивши те що вони мають бути в таблиці, де будуть зберігатись реквізити документу (ів).

Вихідні (похідні) документи системи

$$D_2 = \{\text{"Звіт"}\} = \{d_3\}$$

$$d_3 = x_1 \triangleright \triangleleft x_2 \triangleright \triangleleft x_3$$

$$sh(d_3) = \{a_1, a_2, a_3, a_4, a_4', a_5, a_6\} = \{a_1, a_2, a_3, a_4, a_4', a_5, a_4 / a_4'\}$$

$$shr(d_3) = \{a_1, a_2, a_3\}$$

Спочатку описуються інтуїтивно зрозумілі та необхідні показники. Після цього вони деталізуються, тобто отримуються нові показники. Далі описуються вхідні та вихідні документи.

Документи – це також показники. Різне позначення вхідних показників та документів необхідне для наочного відділення їх, але оперувати ними можна однаково.

Відповідно до п.п 4.4. технологи уточнюють технологічну БЗ. В даному простому прикладі нічого уточнювати не потрібно. Після цього автоматично генерується БД та ППЗ.

Повний алгоритм побудови БД та ППЗ виходить за межі цієї статті. Тут лише приведемо виведення за допомогою вищеописаних правил.

За правилом (6) отримаємо:

$$sh(x_1) = sh(x_2) \Rightarrow \exists(T = x_1 \cup x_2) \Rightarrow sh(T) = \{a_1, a_2, a_3, a_4, a_5\}$$

За правилом (5) отримаємо:

$$shr(x_1) = shr(x_3) \Rightarrow \exists(T = x_1 \triangleright \triangleleft x_3) \Rightarrow sh(T) = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

$$shr(x_2) = shr(x_3) \Rightarrow \exists(T = x_2 \triangleright \triangleleft x_3) \Rightarrow sh(T) = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

Але, як було зазначено вище, така схема створення БД є найпростішою і потрібно використовувати більш жорсткі правила.

Відповідно до правила (7) можемо записати для показників  $x_1$ ,  $x_3$  та звіту  $d_3$ :

$$shr(x_1) = shr(x_3) \wedge \exists d(x_1 = \pi_{sh(x_1)}(d_3) \wedge x_3 = \pi_{sh(x_3)}(d_3)) \Rightarrow$$

$$\exists T(x_1 = \pi_{sh(x_1)}(T) \wedge x_3 = \pi_{sh(x_3)}(T)) \Rightarrow$$

$$sh(T) = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

Отримали таблицю  $T$ , в якій будуть ті ж самі атрибути, що й при використанні правила 5. Але тут враховано використання показників в документах. Тобто, якщо б не знайшлося жодного документа в якому показники  $x_1$  та  $x_3$  поєднані, то їх значення можна було б зберігати в окремих таблицях та позбутися зайвих проекцій.

## Висновки.

1. Побудова ІСУБ, керованих безпосередньо бізнес-персоналом без залучення ІТ-технологів поліпшує вартісні показники систем при одночасному покращенні якості їх функціонування в результаті глибокої та оперативнішої адаптації інформаційної технології до змін оточення та процесів розвитку.
2. Конструктивною лінгвістичною основою для вирішення задачі може бути поняття показника, оскільки управління бізнесом організується навколо моніторингу певних економічних, технічних та технологічних показників бізнес-процесів.
3. Показники можливо представити, як спеціальні реляційні проекції, що дає можливість застосувати напрацьований апарат еквівалентних перетворень реляційної алгебри для автоматизації рішень на основі технологічної бази знань.
4. Існуючі правила в контексті описаних задач мають бути доповнені запропонованою в статті правилами висновків, щодо структури БД та правилами спрощення виразів алгебри показників.

**Список літератури:** 1. Гарсія-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс. – М.: Вильямс, 2002. – 1088 с. 2. Кодд Э.Ф. Расширение реляционной модели для лучшего отражения семантики // СУБД. –1996. – № 5-6. –С. 163 – 192. 3. Мейер Д. Теория реляционных баз данных. – М.: Мир, 1987. – 608 с. 4. Теленик С.Ф. Логика представления процессов функционирования интеллектуальных систем // Системные технологии. Компьютерная обработка экспериментальных данных: Сб. научн. тр. - Днепропетровск, 1999. – Вып 7. 5. Лузан А.В., Теленик С.Ф., Цокол С.Л., Асельдерова И.М. Языковые средства общения пользователя с экономическими информационными системами // Проблемы развития АСУ и информационных услуг в новых условиях хозяйствования: Тез. докл. – Душанбе: Таджик НИИ НТИ, 1989.- С.64-65. 6. В.И. Скурихин, А.А. Павлов, Э.П. Путилов, С.Н. Гриша. Автоматизированные системы управления гибкими технологиями. – К.: Техника, 1987. – 166 с., ил. – Библиогр.: с. 162–164.

Надійшла до редколегії: 30.01.08